

РЕАЛІЗАЦІЯ ОБЧИСЛЕНЬ АНАЛІТИЧНИХ ВИРАЗІВ У ПРОГРАМУВАННІ.

Розроблено і алгоритмізовано метод обчислень будь-яких аналітичних виразів на етапі виконання програм для ПК, що мають навчаючий та прикладний характер.

У процесі створення і експлуатації сучасних комп'ютерних програм досить часто виникає потреба аналізу та обчислення введеного користувачем того чи іншого аналітичного виразу. Якщо уявити собі, наприклад, деяку абстрактну програму, то вона, як правило, виконує наступні дві функції:

1. **Навчаючу** - учням подається необхідний теоретичний матеріал, що підкріплюється конкретними прикладами і (або) задачами.
2. **Контролюючу** - перевіряється рівень отриманих знань та набутих умінь і навичок.

Розглянемо, для визначеності, поетапне створення навчаючої комп'ютерної програми з теми 11-го класу середньої школи «Відшукування первісних для функцій». Така програма, зокнайперше, повинна надати учневі під час роботи з нею найбільш важливі теоретичні відомості з питань відшукування первісних, а також таблицю первісних для всіх найпростіших функцій. Далі положення теорії демонстративно апробуються на кількох вдалих конкретних прикладах відшукування первісних дещо складніших функцій. І, нарешті, контролююча частина програми має перевірити якість засвоєння учнем поданого матеріалу. Тут, до речі, розроблювач може реалізувати окремою підпрограмою загалом значно більше оптимальної кількості якісно розв'язаних прикладів, а при потребі, ще й урізноманітнити їх за змістом і ступенем складності. Однак «дозвіл ЕОМ» на звертання до цієї підпрограми має бути передбачуваним, тобто обґрунтовано вагомим, нагально потрібним за обставин недорозуміння користувачем суті питання або ж у разі його власного бажання поліпшити свій професіоналізм. Отже, в чомусь така підпрограма виконуватиме **навчаючі** функції, а в чомусь - **контролюючі**.

Реалізація **навчаючої** частини не вимагає докладання надзусиль у програмуванні, тому й не розглядається нами. Що стосується **контролюючої** частини, то тут якраз у програмуванні можуть виникнути чималі труднощі.

Перевірку засвоєння матеріалу можна проводити різними способами. Найчастіше знання перевіряються за допомогою тестів: учню пропонується знайти первісну деякої функції та вибрати з кількох запропонованих відповідей ту, яка, на його думку, є правильною. Але перевірка знань у такий спосіб є досить примітивною, хоча чомусь саме він найчастіше використовується в розробках означеного типу (напевне, через граничну простоту реалізації). Тому варто було б запропонувати дещо інший підхід, який виключає можливість угадування, а саме: тому, хто вчиться, спочатку пропонується знайти первісну для певної функції, після чого він повинен перетворити отриманий аналітичний вираз за деякими правилами в рядок символів (учні в 11-му класі на уроках інформатики ознайомлюються з цими правилами при вивченні мови програмування BASIC) та ввести в ПК. На завершення ПК повинен проаналізувати наданий йому рядок та встановити, чи правильно знайдено первісну.

Реалізувати таким чином **контролюючу** функцію програми було б неважко, якби запис первісної був однозначний. Але на практиці, як відомо, кожен аналітичний вираз може мати безліч представлень. Наприклад,

$$\frac{2 \sin^2 X}{3} \equiv \frac{2}{3} \sin^2 X \equiv \sin^2 X \cdot \frac{120}{180} \equiv \dots$$

Передбачити в програмі всі варіанти запису практично неможливо, тому тут необхідно знайти якийсь евристичний спосіб. Доречно запропонувати, наприклад, наступне: в програмі варто зберігати разом з таблицею функцій, що використовуються в контролюючих завданнях, таблицю відповідних їм первісних (по одній первісній на кожну функцію). Оскільки первісна являє собою теж деяку функцію, то потрібно перевіряти збіг у певній області визначення значень отриманої і поданої учнем первісної з відомою програмі. Щодо практичної сторони цього питання, то можна перевірити збіг їх значень при певній кількості випадкових значень аргументів. У разі збігу цих значень функцій у всіх контрольних точках припустимо з певною точністю стверджувати правильність поданої первісної.

Але тут виникає проблема аналізу та обчислення значень поданої і закладеної в ЕОМ функцій на етапі виконання програми, адже в жодній з відомих мов програмування така можливість не передбачена.

Розглянемо один із способів вирішення цієї проблеми. Для цього, по-перше, спробуємо позначити шлях обчислення комп'ютером деякого аналітичного виразу. Нехай ним буде, наприклад, такий вираз:

$$\frac{\sin^2 12 - 13 \cdot \ln 5}{5,23^{0,34} \cdot 34} \cdot \sqrt{|tg(-20)|}.$$

Представивши останній за відповідними правилами запису аналітичних виразів для мов програмування високого рівня, отримаємо рядок символів, що складається з літер латинського алфавіту та деяких інших допоміжних символів (дужки, крапки, знак множення «*», знак ділення «/», знак піднесення до степеня «^» тощо):

$$(((\text{sqr}(\sin(12)) - 13 * \ln(5)) / (5.23 ^ 0.34 * 34) * \text{sqrt}(\text{abs}(tg(-20)))))) \quad (1)$$

Тепер перед нами стоїть завдання обчислення отриманого рядка символів. Розв'язання цієї задачі можна умовно розділити на кілька етапів.

1. **Первинна обробка поданого запису аналітичного виразу.**

Цей етап включає в себе вилучення пропусків із заданого рядка (ущільнення запису) та попередню перевірку на наявність синтаксичних помилок у цьому рядку (недостача закриваючої або відкриваючої дужки, недопустимі символи в аналітичному записі, помилковий запис числової константи тощо). Вилучивши пропуски, ми перетворимо рядок (1) на наступний вигляд:

$$\langle\langle(\text{sqr}(\sin(12))-13*\ln(5))/(5.23^0.34*34)*\text{sqrt}(\text{abs}(\text{tg}(-20)))\rangle\rangle \quad (1.1)$$

Оскільки отриманий запис цілком коректний з точки зору подальших операцій, то далі переходимо до наступного етапу.

2. Початковий аналіз та синтаксичні перетворення поданого запису.

Для спрощення подальшої обробки виразу, на цьому етапі проводиться пошук у рядку записів відомих функцій (послідовність літер латинського алфавіту типу «ARCTG» (функція, що повертає арктангенс аргументу) чи «SQRT» (функція, що повертає корінь квадратний з аргументу)) та заміна їх на відповідні позначення. З цим на початку роботи кожному запису функції ставиться у відповідність деякий унікальний символ, у ролі якого зручно використати прописні літери латинського алфавіту. Так, наприклад, у розроблюваній програмі вибрано наступну інтерпретацію:

Назва	Запис	Символ
арктангенс	«ARCTG»	a
синус	«SIN»	b
косинус	«COS»	c
котангенс	«CTG»	d
тангенс	«TG»	e
експонента	«EXP»	f
дробова частина	«FRAC»	g
ціла частина	«INT»	h
логарифм натуральний	«LN»	i
корінь квадратний	«SQRT»	j
квадрат	«SQR»	k
модуль	«ABS»	l

Після виконання даної операції рядок (1.1) матиме вигляд:

$$\langle\langle(k(b(12))-13*i(5))/(5.23^0.34*34)*j(l(e(-20)))\rangle\rangle \quad (2)$$

Далі на цьому ж етапі потрібно запропонувати збереження у деякому масиві присутніх у поданому рядку числових констант та заміну записів цих констант на вказівки до відповідних елементів даного масиву. Для цього необхідно перед початком обробки зарезервувати такий масив і, для зручності, вибрати для нього індексацію через великі літери латинського алфавіту. У нашій програмі цей масив має назву Number і складається з 26 комірок для дійсних чисел, що індексовані літерами від «A» до «Z». Це, звичайно, накладає деякі обмеження на використання числових констант у записі аналітичного виразу (в даному випадку - до 26 констант), але в загальному випадку такий розмір масиву цілком достатній для обчислення практично любых виразів.

Повернемося тепер до отриманого рядка (2). Легко помітити, що він містить лише сім записів констант. Це числа 12, 13, 5, 5.23, 0.34, 34 та 20. Зберігши ці константи в перших семи елементах масиву Number, що проіндексовані від «A» до «G», та замінивши їх запис у рядку (2) на вказівки до відповідних елементів масиву, отримаємо наступний рядок:

$$\langle\langle(k(b(A))-B*i(C))/(D^E*F)*j(l(e(-G)))\rangle\rangle \quad (2.1)$$

Після цього масив Number міститиме константи:

1.1. Комірка	A	B	C	D	E	F	G	...
Вміст	12	13	5	5.23	0.34	34	20	...

Порівнявши початковий запис (1) з отриманим (2.1), помічаємо очевидне суттєве спрощення поданого рядка.

До цього моменту ми лише спрощували, абстрагували поданий запис, не обчислюючи його. Тому в наступному пункті перейдемо до обчислень.

3. Безпосереднє обчислення поданого запису.

На даному етапі відбувається послідовне виділення та обчислення виразів у дужках, починаючи з перших внутрішніх дужок і закінчуючи зовнішніми, керуючись правилами пріоритету операцій. Так, першочергово виконуються операції піднесення до степеня, потім обчислення значення функції, далі йде виконання операцій множення та ділення, і вже потім операції додавання та віднімання.

Програма, що має реалізувати описані вище дії, повинна містити щонайменше два блоки: блок відшукування та виділення «найглибших» виразів у дужках і блок безпосереднього обчислення виділених першим блоком виразів. У програмі перший блок реалізовано в тілі основної підпрограми Calculate у вигляді циклу, що відповідає за знаходження відповідних виразів у дужках, передачі цих виразів до другого блоку та маніпуляції з поверненим другим блоком результату, а блок обчислення винесено в окрему локальну підпрограму Inside, що отримує у вигляді аргументу знайдений першим блоком вираз у дужках, обчислює його і повертає першому блоку вказівку на елемент масиву Number, в якому було збережено результат обчислень.

Зупинимося більш детально на реалізації підпрограми Inside. В ній здійснюється триетапний аналіз отриманого як аргумент виразу, де на першому етапі відбувається видалення символів дужок, пошук символу «^» (піднесення до степеня) та виконання відповідної операції, на другому етапі - пошук малих латинських літер (скорочені позначення записів функцій) та обчислення відповідних функцій, на третьому етапі - пошук символів «*», «/», «+», «-» та виконання відповідних операцій (спочатку всі операції ділення та множення, потім додавання та віднімання).

У даному випадку у рядку (2.1) міститься дев'ять виразів у дужках, «найглибшим» з яких є «(A)». Цей вираз не містить символів ніяких математичних операцій. Тому блок обчислень поверне як результат той самий вираз, знищивши лише символи дужок (до речі, символи дужок необхідно знищити, щоб не відбулося зациклення першого блоку). Блок виділення виразів у дужках, у свою чергу, повинен видалити з рядка символи «(A)», а на їх місце помістити отриманий від блоку обчислень результат - символ «A». Після такого перетворення рядок (2.1) дещо зміниться:

$$\langle\langle(k(bA)-B*i(C))/(D^E*F)*j(l(e(-G))))\rangle\rangle \quad (3)$$

Тепер «найглибшим» є вираз «(bA)». Аналогічно, перший блок передає другому цей вираз для обчислення. Тут так само знищуються символи дужок і проводиться обчислення функції \sin з аргументом, що міститься в комірці A масиву Number (тобто обчислюється $\sin 12$), після чого результат зберігається в тій самій комірці. Далі в перший блок повертається результат «A», після чого з рядка (3) будуть видалені символи «(bA)», а на їх місце вставлено символ «A». Отримаємо:

$$\langle\langle(kA)-B*i(C))/(D^E*F)*j(l(e(-G))))\rangle\rangle \quad (3.1)$$

Відповідно зміниться вміст масиву Number:

Комірка	A	B	C	D	E	F	G	...
Вміст	-0.53657	13	5	5.23	0.34	34	20	...

У результаті послідовного виконання описаних дій відбуватиметься спрощення поданого рядка і, відповідно, змінюватиметься вміст масиву Number.

$$\langle\langle(kA)-B*i(C))/(D^E*F)*j(l(e(-G))))\rangle\rangle \quad (3.2)$$

Тут поки що вміст масиву Number не змінився.

$$\langle\langle(A/(D^E*F)*j(l(e(-G))))\rangle\rangle \quad (3.3)$$

Комірка	A	B	C	D	E	F	G	...
Вміст	-20.6347	20.9227	1.6094	5.23	0.34	34	20	...

$$\langle\langle(A/E*j(l(e(-G))))\rangle\rangle \quad (3.4)$$

Комірка	A	B	C	D	E	F	G	...
Вміст	-20.6347	20.9227	1.6094	5.23	59.6718	34	20	...

$$\langle\langle(A/E*j(l(eG))))\rangle\rangle \quad (3.5)$$

Комірка	A	B	C	D	E	F	G	...
Вміст	-20.6347	20.9227	1.6094	5.23	0.34	34	-20	...

$$\langle (A/E*j(IG)) \rangle \quad (3.6)$$

Комірка	A	B	C	D	E	F	G	...
Вміст	-20.6347	20.9227	1.6094	5.23	0.34	34	-2.2372	...

$$\langle (A/E*jG) \rangle \quad (3.7)$$

Комірка	A	B	C	D	E	F	G	...
Вміст	-20.6347	20.9227	1.6094	5.23	0.34	34	2.2372	...

$$\langle A \rangle \quad (3.8)$$

Комірка	A	B	C	D	E	F	G	...
Вміст	-0.5172	20.9227	1.6094	5.23	0.34	34	1.4957	...

Отже, в результаті такого ланцюжка операцій рядок (3) перетворився в рядок (3.8). По суті, рядок (3.8) являє собою вказівку на комірку масиву Number, що містить результат обчислення всього виразу. В даному випадку ми отримали результат у комірці A, і він становить -0.5172. Таким чином, процес обчислення виразу потрібно припинити тоді і тільки тоді, коли в рядку залишиться лише один з них - вказівка на комірку масиву, що містить результат обчислень.

Вітюк Олександр Володимирович - студент фізико-математичного факультету Житомирського державного педагогічного інституту ім. І.Франка.

Наукові інтереси:

- інформатика;
- методика викладання інформатики і математики.

Ленчук Іван Григорович - кандидат технічних наук, доцент, завідувач кафедрою математики Житомирського державного педагогічного інституту ім. І.Франка.

Наукові інтереси:

- системи автоматизованого проектування;
- прикладна і конструктивна геометрія.